**FIG. 1**

202	210	200	206	208
	<u>Why is tel broken?</u>	204	(53)	06/10/99 – 06/22/99
(Mira) Hello. I'm trying to collect a list of all the <u>_current_</u> issues making tel a bad language choice. I'd like concrete examples, and not just vague assertions that ...				
	<u>MAC-address</u>		(1)	06/10/99
(Jannsen) High, is there a way to get the MAC-addresses and other properties of network cards with python? thanx. Holger ...				
	<u>Asserting object class</u>		(4)	06/10/99
(Adi) In a method, I'd like to asset that a parameter is an object which is of a specific class or has this class as it's base class. I can test on the <u>_class_</u> and its ...				
	<u>KBHIT in a Windows Environment</u>		(2)	06/10/99
(Zimmerle) I am using Python 1.52 in a Windows 98 environment. I recently tried using the kbhit() function in the msvert module. It appears to work fine in the Python ...				
	<u>Pythonwin – advise sought</u>		(2)	06/10/99
(Williams) I am looking at Pythonwin as an alternative to Tk and would appreciate advice on the following three questions. The Pythonwin documentation points to MFC ...				
	<u>force sequenz with only one item?</u>		(4)	06/10/99 – 06/14/99
(Jannsen) High, I thin it's easy to solve, but I couldn't get it right now. Take a look at the command-line source below: >>> A = 0 >>> B = 1 >>> seq=raw_input("?") ?A,B>>> ...				
	<u>Date formats</u>		(4)	06/10/99 – 06/11/99
(Milliger) I read the documentation in the Python Reference Library (6.3) on time access and conversions and chapter 8 in programming python, yet I can't figure out if ...				
	<u>Advice on an how to handle errors in extension module.</u>		(1)	06/10/99
(Oliphant) I'm writing some Python-C interfaces to some mathematical libraries in FORTRAN. In particular, I'm working on interfacing QUADPACK with Python. The code is ...				

FIG. 2

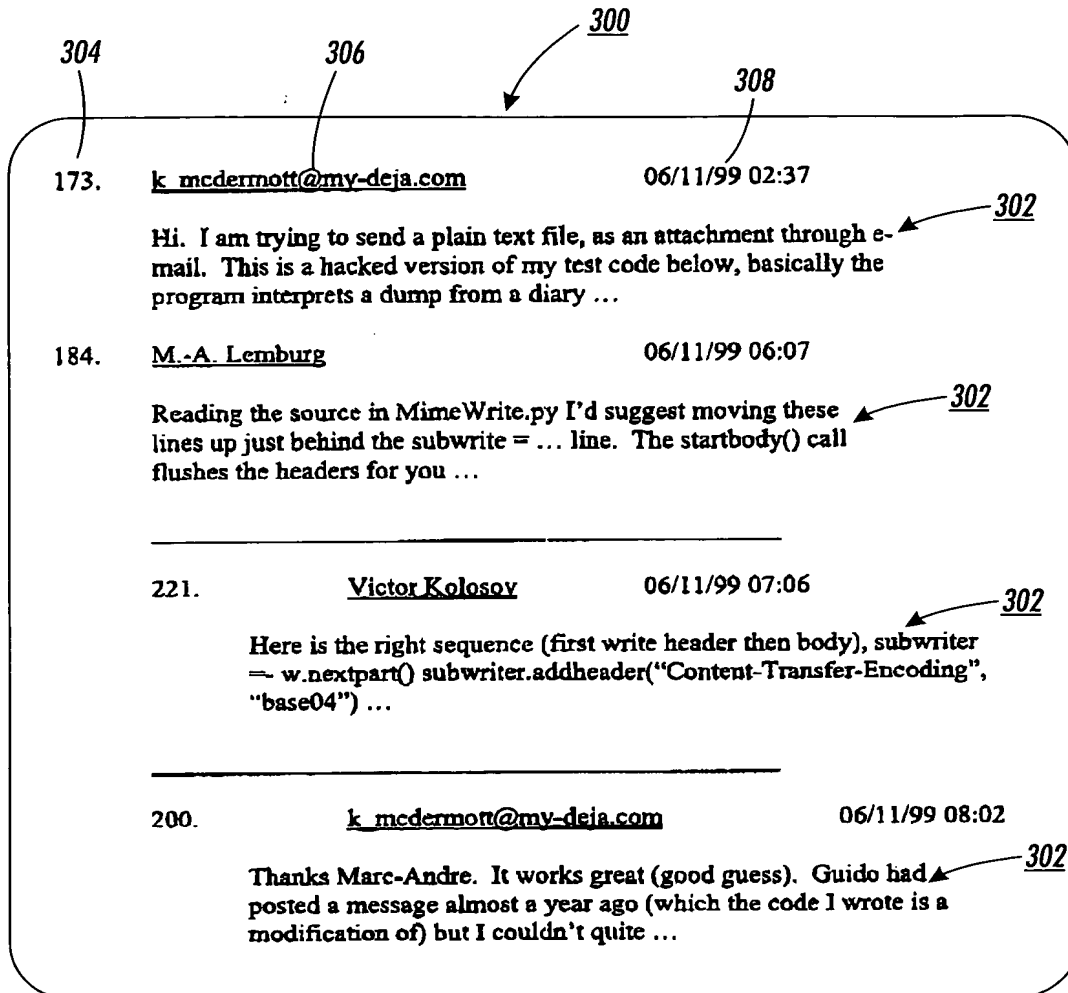


FIG. 3

723. Mark Hammond

06/18/99 17:15

[Schulenburg: I have been using a freeware product called AutoDuck to document my Python source code.]

Cool! I use AutoDuck to document all my Python extension C++ code. All the helpfiles in the Win32 extensions are generated using autodock.

Sorry in advance for the length of this, but I am in exactly the same predicament as you, and have been thinking for a number of years how to resolve this. I'm really pleased to finally find someone else with this problem, and really hope we can solve it in a reasonable way. ...

However, depending on how far you have gone with AutoDuck, pythondoc/gendoc isn't solving the exact same problem.

IMO, AutoDuck generates far superior reference documentation.

A new release of pythondoc uses XML. I believe, so PythonDuck could be a tool that can enhance the documentation generated by pythondoc.

But in summary, Pythondoc exists and is very cool, but doesn't quite cover what AutoDuck can do (and vise-versa, but that isn't the point now). I believe a real need exists for the functionality AutoDuck provides, and further I need it personally!

FIG. 4

723. Mark Hammond

06/18/99 17:15

[Schulenburg: I have been using a freeware product called Autoduck to document my Python source code.]

Cool! I use AutoDuck to document all my Python extension C++ code. All the helpfiles in the Win32 extensions are generated using autoduck.

As another poster pointed out, you should look at pythondoc/gendoc. Sorry in advance for the length of this, but I am in exactly the same predicament as you, and have been thinking for a number of years how to resolve this. I'm really pleased to finally find someone else with this problem, and really hope we can solve it in a reasonable way.

However, depending on how far you have gone with AutoDuck, pythondoc/gendoc isn't solving the exact same problem. Well, they all attempt to solve the same problem, but have different strengths.

IMO, AutoDuck generates far superior reference documentation. Pythondoc's big strength is the integration with Python.

AutoDuck as 3 strengths over pythondoc that I am having trouble reconciling: pythondoc really doesn't support natural links to objects. Autoduck can say "see method" or "@xref object.method" to generate inline links or a cross-references section. Has been some discussion how to expand Pythondoc to support this, but a) it hasn't been done and b) it still is not as natural as autoduck.

I have come to the conclusion that we need a _new_tool - PythonDuck. It should work with Pythondoc, and be capable of enhancing its output. A new release of pythondoc uses XML. I believe, so PythonDuck could be a tool that can enhance the documentation generated by pythondoc. Alternatively we just graft it in, and beat up Daniel until he agrees :-)

This tool could possibly still work from comments, or maybe from the tad-end of the docstrings (thereby stripped from the documentation, and not messing the raw docstring too much). #@ comments could be considered "directives" for the generated output. Eg. "#@ xref SomeOtherFunction" will create a cross-references section. A comment such as "#@ paramtype param_name" would direct that the param called "param_name" should be documented as being of type "SomeType" and a cross-reference to the SomeType definition be added. The tool should resolve all links at the end, and report bad links.

I know I am not making my point that well, and unfortunately don't have time to fix this. But in summary, Pythondoc exists and is very cool, but doesn't quite cover what AutoDuck can do (and vise-versa, but that isn't the point now). I believe a real need exists for the functionality AutoDuck provides, and further I need it personally! So we should try and make this happen.

Mark.

FIG. 5

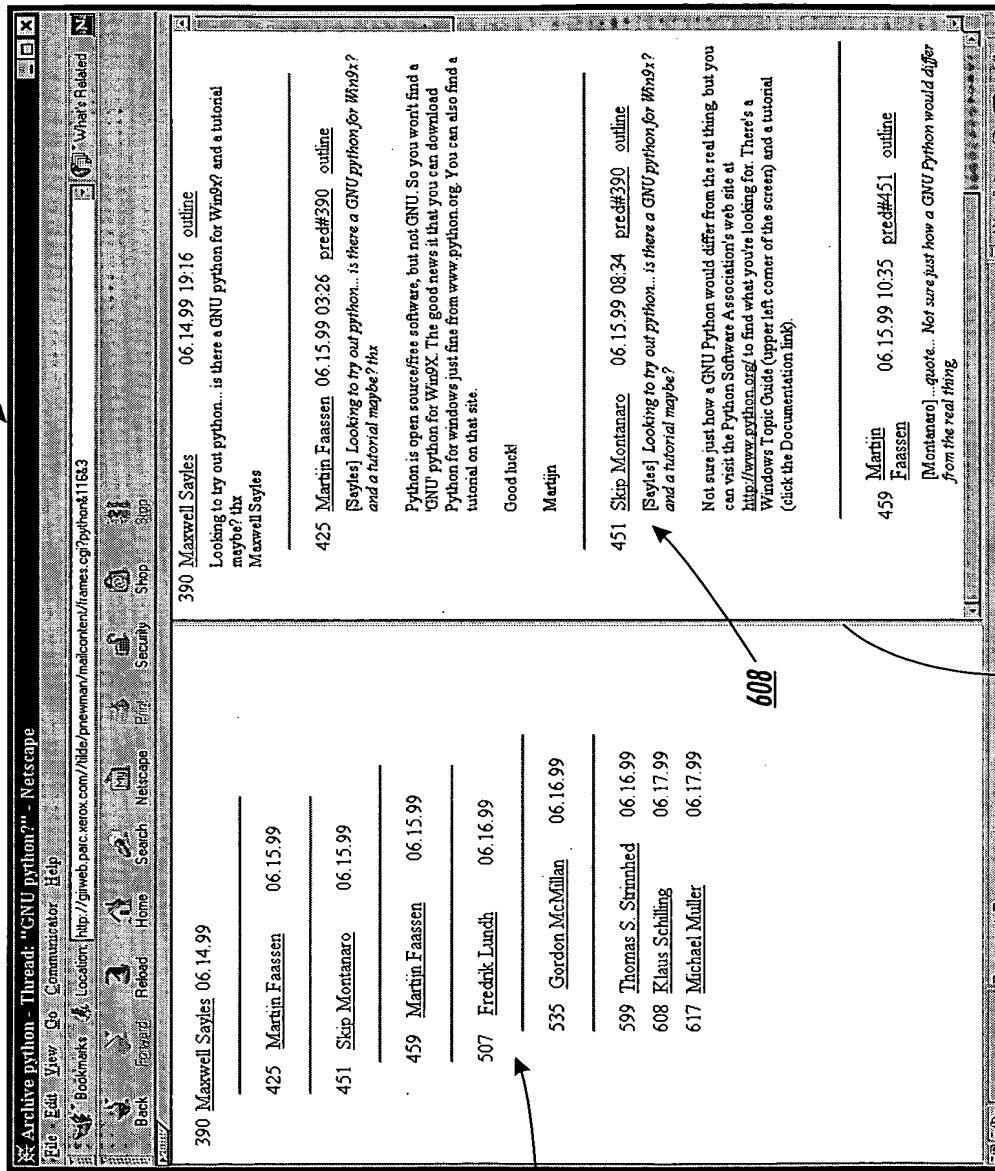


FIG. 6

700

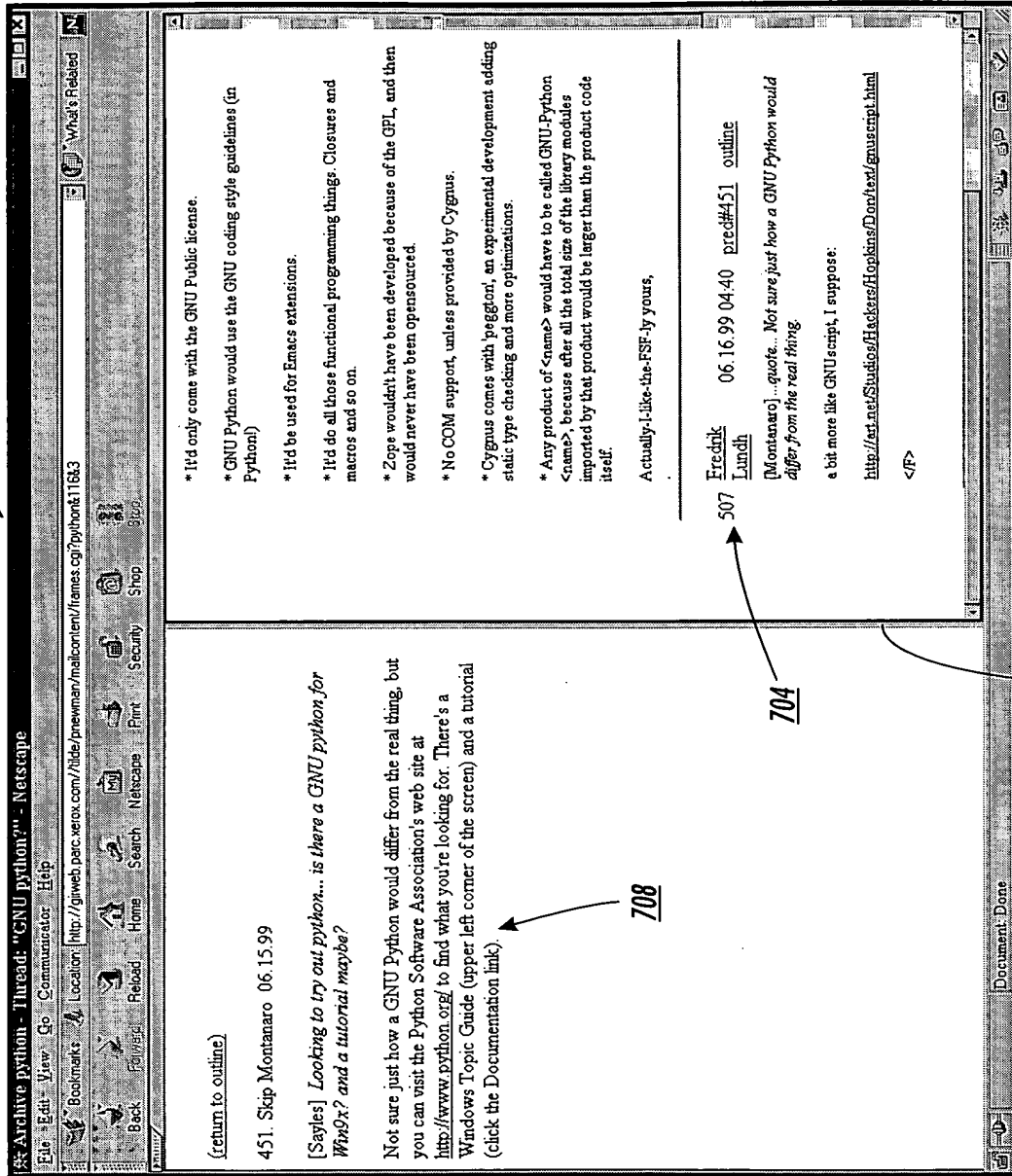


FIG. 7

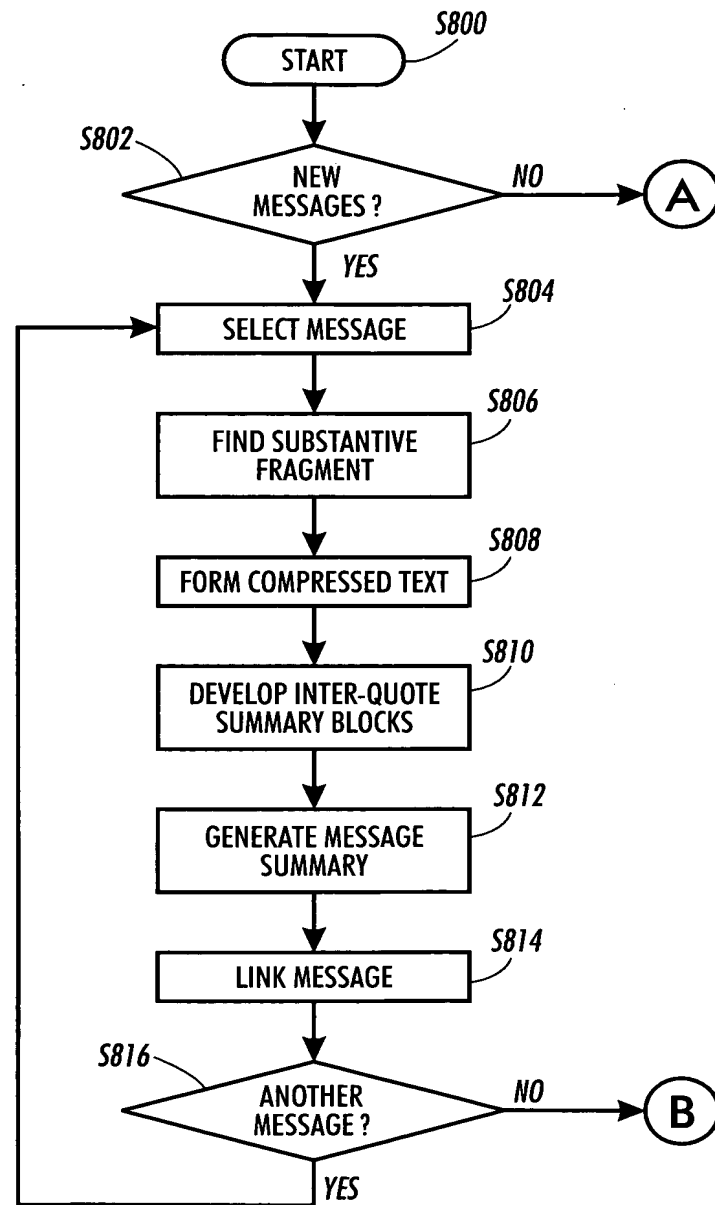


FIG. 8

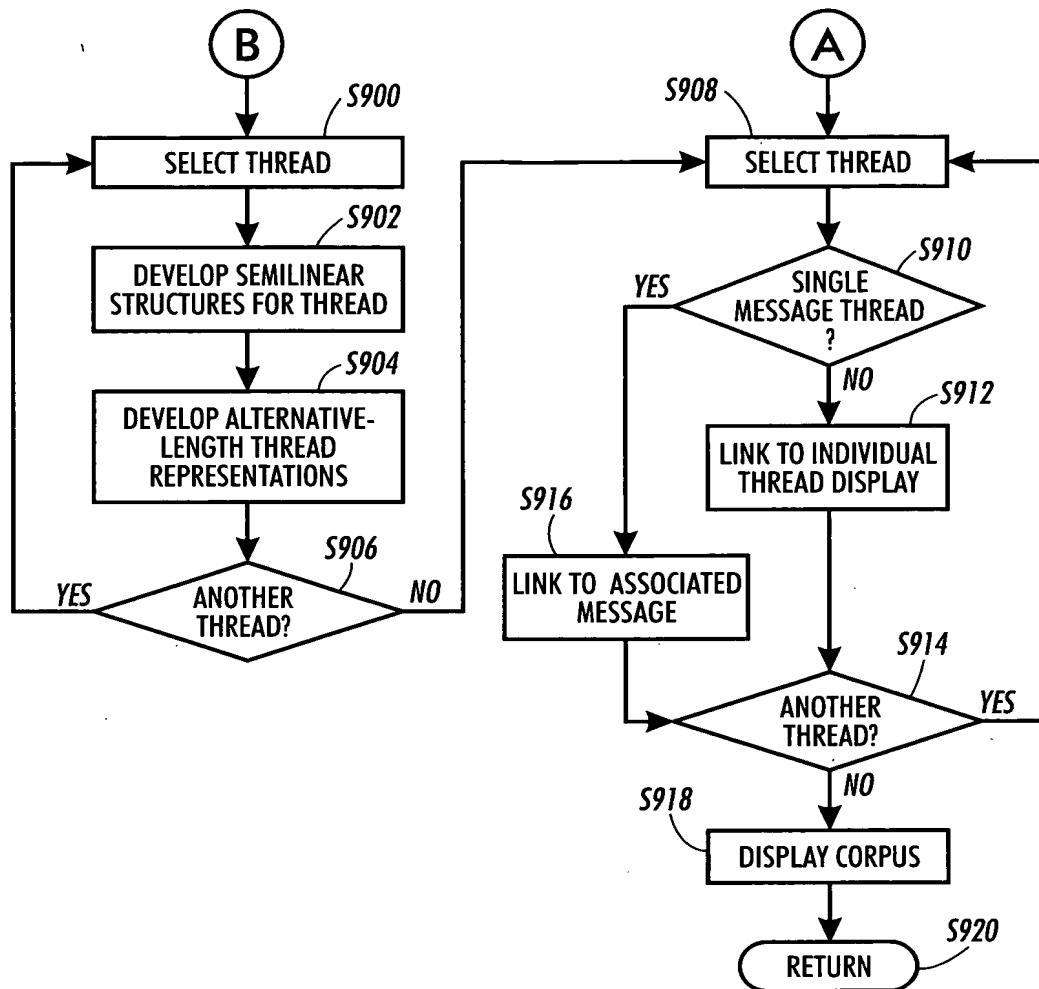


FIG. 9